



The
Patent
Office

REC'D 03 APR. 2000



INVESTOR IN PEOPLE

IPO

PCT

GB00/00762

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

EU

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

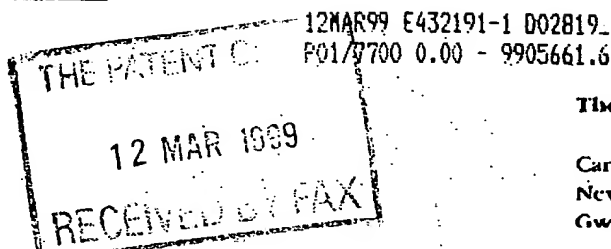
Dated 20 MAR 2000

BEST AVAILABLE COPY

Patents Form 1/77

Patents Act 1977
(Rule 16)The
Patent
Office

Request for grant of a patent

(See the notes on the back of this form. You can also get
an explanatory leaflet from the Patent Office to help
you fill in this form)

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference

C882/A

2. Patent application number

(The Patent Office will fill in this part)

12 MAR 1999

9905661.6

3. Full name, address and postcode of the or of
each applicant (underline all surnames)Advanced Rendering Technology Limited
Mount Pleasant House
2 Mount Pleasant
Huntingdon Road
Cambridge CB3 0BL
United Kingdom

Patents ADP number (if you know it)

If the applicant is a corporate body, give the
country/state of its incorporation

7207590001

4. Title of the invention

Method for Convolution of Signal or Image Data

5. Name of your agent (if you have one)

Keith W Nash & Co

"Address for service" in the United Kingdom
to which all correspondence should be sent
(including the postcode)90-92 Regent Street
Cambridge
CB2 1DP

Patents ADP number (if you know it)

1206001 ✓

6. If you are declaring priority from one or more
earlier patent applications, give the country
and the date of filing of the or of each of these
earlier applications and (if you know it) the or
each application number

Country

Priority application number
(if you know it)Date of filing
(day / month / year)7. If this application is divided or otherwise
derived from an earlier UK application,
give the number and the filing date of
the earlier application

Number of earlier application

Date of filing
(day / month / year)8. Is a statement of inventorship and of right
to grant of a patent required in support of
this request? (Answer "Yes" if:

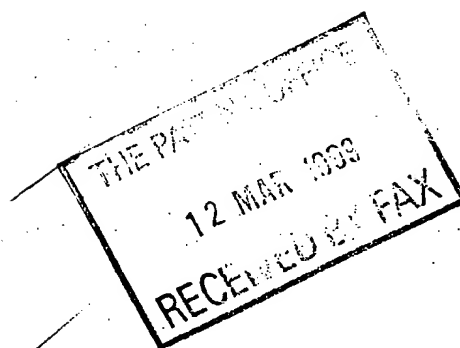
- a) any applicant named in part 3 is not an inventor, or a) Yes
b) there is an inventor who is not named as an
applicant, or
c) any named applicant is a corporate body.
See note (d))

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description	25
Claim(s)	-
Abstract	-
Drawing(s)	4



10. If you are also filing any of the following, state how many against each item.

Priority documents

Translations of priority documents

Statement of inventorship and right to grant of a patent (Patents Form 7/77)

Request for preliminary examination and search (Patents Form 9/77)

Request for substantive examination (Patents Form 10/77)

Any other documents (please specify)

11. I/We request the grant of a patent on the basis of this application.

Signature Keith W. Nash & Co. Date 12.03.99

Keith W Nash & Co, Agents

12. Name and daytime telephone number of person to contact in the United Kingdom Mr. D. L. Roberts 01223 355477

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

- If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.
- Write your answers in capital letters using black ink or you may type them.
- If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.
- If you have answered 'Yes' Patents Form 7/77 will need to be filed.
- Once you have filled in the form you must remember to sign and date it.
- For details of the fee and ways to pay please contact the Patent Office.

12-MAR-99 FRI 14:25

KEI NASH & Co.

FAX NO. 012 324353

P.05

1

C882/A

TITLE: Method for Convolution of Signal or Image Data

Field of the Invention

This invention relates to a method of convolving an input data set with a further data set to produce an output data set. The method is particularly, but not exclusively, applicable to image processing in which a data set constituting an image is convolved with a further data set (which may act as a filter).

Background to the Invention

Convolution is a useful computational technique, which is used in many signal and/or image processing applications. For all combinations of two data elements, one from each set, the products of the values of the two data elements are found and the positions of the elements in the two sets are used to map the resultant data into an output data set. The technique may be used, for example, to simulate certain photographic effects in a computer generated image, for example lens blooming and internal reflections.

Certain methods directly find the products of all possible combinations of data from the two sets. These methods can be very slow particularly for large data sets, as would generally be involved with image processing.

The convolution process can be accelerated using Fourier Transform methods by which the data sets are transformed into the frequency domain and then convolved, for example by the Fast Fourier Transform (FFT) method. However, such techniques can still nevertheless be time consuming, and can require a considerable amount of temporary data storage to be available

in the apparatus performing the convolution.

It has also been found that even if one of the data input sets is sparse (i.e. has lots of zero-valued elements or elements with very small values) either of the above techniques can still be time consuming.

Background to the Invention

According to the present invention, there is provided a method of convolving an input data set with a further data set, each of the sets containing a plurality of data elements, performing convolution calculations mathematically to combine the values of the data elements of the two sets and to map the results into an output data set, the method comprising the steps of:

(a) Arranging the elements of each of the input and further data sets into a respective set of groups;

(b) For each group, obtaining an average of the values of the data elements contained in that group and obtaining at least one variance value representative of the range of values of those elements;

(c) Determining for which, if any, pair combination of groups, one from each set, the results of convolution calculations based on average values for at least one of the groups in the pair would not differ significantly from the results of convolution calculations based on the individual values of the data elements contained in both groups of the pair, and using the average value or values in the convolution calculations for the combinations of data elements for those pairs by treating the values of all the data elements in the or both groups as being constant and substantially equivalent to the average value for their group;

(d) Basing the convolution calculations for the remaining

pair of combinations of groups, if any, on the individual values of data elements contained therein.

Thus, the invention determines whether the elements contained in a given group can be represented as having the same value for the purposes of the convolution calculations. If they can, this reduces the amount of computation need to perform the calculations since the individual data values from the group concerned do not have to be read at least for those calculations. Furthermore, if both groups in a given pair can be represented by average values, the product value only has to be calculated once in the convolution of the data from those two groups.

The invention thus provides a way of finding suitable approximations for values of certain data elements in order to reduce the amount of data which has to be processed in the convolution calculations. The invention therefore achieves an approximate convolution to an acceptable level of accuracy.

Preferably, the input data set comprises a representation of a perceivable subject and may thus, for example, form at least part of an image or an audio signal. However, the invention is particularly applicable to an input data set representative of an image (each data element corresponding to a respective image pixel, with the value of that element corresponding to colour or brightness of the pixel). Thus, the invention can provide a method of processing an image signal by convolving the image data carried thereby with a further data set.

Preferably, each set of groups is hierarchically arranged, with the members of each group at the bottom level of each hierarchical set being constituted by the data elements of that set, and the members of the or each group in the or each higher level being constituted by groups from the level below, so that, the higher a group in the hierarchy, the larger the number of data elements it contains (either directly or through

12-MAR-99 FRI 14:26

KEITH H. NASH & Co.

FAX NO. 01223 324353

P.08

4

containment of other groups from lower levels).

Preferably, in step (c) the groups of each hierarchical set are considered in descending order so that any given pair of groups is analysed in accordance with step (c) only if it has been determined that average values for the higher level groups, which contain the members of that given pair, are not to be used.

This provides an effective way of determining whether average values can be used in respect of large proportions of the data sets at a relatively early stage of the convolution method.

If the input data set constitutes an image, the groups can correspond to regions and sub-regions (each contained in a region) of the image so that the higher the group in the hierarchy the larger the region it represents.

Said representative variance values for any given group may comprise the difference between the maximum and minimum values of the data elements contained therein, in which case it can be determined whether the elements contained in the group can all be treated as being of the same value, equivalent to their average, for the convolution calculations if the difference is less than a given threshold.

Preferably, however, the representative variance values for each group comprise the maximum and minimum values of data elements contained thereby, and the elements of each group in any group pair combination are all treated as having the same value (i.e. the average value) if the difference between the products of the maximum values and the products of the minimum values of the two groups is below a given threshold.

In the case of image data, the threshold may be that value below which there would be no perceivable variation in the portion of the output data created by convolving the two

THIS PAGE BLANK (USPTO)

groups.

Preferably, for image data, the method includes the step of modifying said threshold, after the data sets have been partially convolved, independent on the portions of the output data set already created by the partial convolution. For example, the threshold may be raised where the output for a given group is to be situated adjacent a portion of the output image which has already been determined and is relatively bright, or is of maximum brightness.

Preferably, for each pair combination of sets for which it has been determined that convolution calculations based on average values would not differ significantly from calculations based on the individual values of the data elements contained in those groups, the method includes a further step of treating the value of all the elements in both groups as zero if the products of the averages for the two groups is below a given threshold.

Preferably, this threshold can also be modified in response to output data already obtained in a similar fashion to that in which the threshold which is applied in relation to the variance values can be modified.

If only a region of the output data set is required, the method preferably includes the additional step of determining which of the groups contain data elements, the convolution calculations on which will produce a result outside said region in the output data set, and discarding those groups.

Preferably, this step is performed before step (c) above.

It will be appreciated that the method of the present invention can significantly expedite convolution of two image data sets where at least one of the sets has relatively large areas of uniform brightness/colour (depending upon the values being

convolved) or large areas of zero intensity). Also, the discarding of certain groups as discussed above can significantly reduce the amount of processing required for the convolution, and is a step which is not possible where Fourier methods are used, since the latter require the transformation into the frequency domain of both the input and further data sets in their entirety.

Brief Description of the Drawings

The invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a schematic diagram showing two one-dimensional data sets (in simplified form) which may be convolved by a method in accordance with the invention;

Figure 2 is a schematic diagram showing how two other one-dimensional data sets (which are to be convolved in accordance with the present invention) can be arranged into two hierarchical sets of groups;

Figure 3 illustrates the contribution made by the convolution of selected portions of the two data sets in Figure 2 to the output data set for the convolution;

Figure 4 illustrates data which is stored in respect of each of the groups illustrated in Figure 2; and

Figure 5 is a diagram (similar to Figure 1) showing three data sets, one having been formed by the convolution of the other two.

Detailed Description

Figure 1 shows two data sets, represented as two columns of boxes. One of those columns, column generally referenced 1,

represents an input data set for a line of an image. Each individual box in the column represents a pixel in that line, and the numerical value contained in the box is representative of the brightness intensity level of the pixel, whilst the position of the box in the data set is related to the position of the pixel in the image line. It will be appreciated that the data set has been greatly simplified by only showing a relatively small number of pixels. In reality, the line of the image would contain several hundred pixels, and a column with a corresponding number of boxes would be needed to represent such a line.

Column 2 represents a further data set which acts as a kernel of the convolution to be performed on the input data set. Like the input data set 1, the kernel corresponds to a line of an image of a "filter" to be applied to the data set 1. The numbers inside the boxes in the column of the data set 2 therefore correspond to pixel intensities, as is the case with the input data set. However, the relative position of each box in the further data set 2 provides a displacement function which will be discussed in more detail in relation to Figures 2 and 3.

As with the input data set 1, the data set 2 has been greatly simplified and would, in reality, have a much larger number of individual elements.

The convolution function involves obtaining the products of each data element of the input data set 1 with all the data elements of the data set 2, mapping those products into positions in an output data set. If a plurality of products are mapped into the same position in the output set, their sum is also found. With a large number of data elements in both the sets, performing these calculations in respect of all the individual data element pair combinations would require a very large number of individual calculations to be performed. However, the convolution process can be considerably expedited

8

by using an approximation of the data element intensities whereby certain groups of data elements are treated as having the same intensity, equivalent to a representative "average" value.

Accordingly, the method of the present invention commences by sorting the data elements into a hierarchical structure of groups, and Appendix 2 is a pseudo program code for constructing such a hierarchy.

Figure 2 shows an input data set 4 and a further "filter" data set 6, both of which constitute single lines in a respective image. These data sets are represented as rows of boxes, each of which is a data element corresponding to a respective pixel. The numbers appearing in the boxes represent the intensities of the corresponding pixels, whilst the numbers set out beneath the boxes (Nos 0-15 for set 4 and -7 to 8 for set 6) are the positional data for the individual data elements. In reality, a far larger number of pixels, and hence data elements, would be present.

The individual data elements of the set 4 are arranged into a hierarchical set of groups which is depicted by means of a binary-tree generally referenced 8, whilst a similar binary-tree generally referenced 9 depicts the grouping of the data elements of the set 6 into a hierarchical group structure.

Each node of each tree represents a respective group, and the connections between the groups indicate which groups contain which. The leaf nodes of the tree (i.e. the bottom of the tree) represent the individual data elements. Thus, the individual data elements are at level 4 of the hierarchy, and each of the groups in level 3 contain a respective pair of data elements. For example, the node 10 represents a group which contains data elements at positions 0 and 1 in the set 4. The members of each node in level 2 comprise a respective pair of adjacent nodes from level 3, so that the nodes at level 2 each

contain a respective set of four consecutive data elements.

Thus, the group depicted by the node 12 contains the groups 14 and 16 from level 3, and thus also the data elements at positions 4-7 inclusive.

It can be seen from the diagram that each group at level 2 contains pixels which correspond to a respective region constituting a quarter of the image line. The group represented by each node at level 1 contains a respective pair of groups represented by the nodes at level 2, and therefore corresponds to half of the image line. Thus, for example, the group represented by the node 18 contains those groups represented by the nodes 12 and 20. The tree 8 has a root node 22 which corresponds to the whole of the image line. Similarly, each group at level 3 of the structure represented by the tree 9 contains the elements for a respective eighth of the image, a respective quarter of the image is contained in each group at level 2, half the image in each group at level 1 and the whole filter image at the group represented by root node 24.

For each node, there is obtained the following values which are representative of the data elements contained by the group represented by the node:

1. The average value of the intensity of the elements (in this example the average is the mean average and the average values for the nodes at levels 0, 1 and 2 are obtained by taking the mean average values of the relevant nodes from the level below);

2. The intensity value of the data elements with the maximum intensity value;

3. A minimum intensity value equal to that of the element with the minimum intensity;

12-MAR-99 FRI 14:28

KEITH NASH & Co.

FAX NO. 012 324353

P.14

10

4. The start position of the data elements (the position of the left-hand data element in this example);

5. The length of the region represented by the data elements.

The average intensity values have been written into the boxes indicating individual nodes in the tree shown in the Figures. Thus, for example, the average intensity associated with the node 20 is .1, the maximum intensity is .2, the minimum 0, the node start position is 0 and the node length is 4. The data stored for each node are represented in Figure 4 and also includes data identifying the child nodes (or sub-nodes) i.e. those nodes (or data elements if the node is in level 3) from the level below to which the node is connected. The maximum and minimum values stored in respect of the nodes constitute the representative variance values which are used to provide a measure of the variation in the values which the convolving of the data elements concerned would contribute to the output. Depending on the use of the output image, a threshold value, referred to as TV, can be calculated such that TV is the largest change in intensity of picture elements in the output image to which a human viewer is insensitive when the image is represented on an output display.

The convolution of any given pair combination of individual data elements of the sets 4 and 6 involves taking the products of the intensities of those elements, and mapping that product into an output data domain part of which is shown at 26 in Figure 3. The address of the output domain into which the product is mapped is determined by taking the position of the relevant data element from the set 4 and displacing it by an amount determined by the position data for the element in the set 6. For example, the convolution of the data elements 28 (from set 4) and 30 (from set 6) will give a product of $10 \times 1 = 10$, and this will contribute to the value at the address 5 (i.e. 5×7) in the set 26. The final value at the address

5 of the set 26 will be the sum of all the products which are mapped to that address. The product contribution for the elements 28 and 30 is indicated by the notation $I(4,5)*F(4,7)$, and a similar notation is used to indicate the contributions of other node pairs in the data sets. The notation can be generally represented as $I(a,b)*F(c,d)$ where:

I indicates the input data set 4,

F indicates the filter data set 6,

a and c represent the respective levels (0-4) of the node, and

b and d represent the number of places (from 0 to the total number of nodes at the level -1) from which the node is spaced from the left side of the level.

Some products will be mapped into other addresses in the data set, but these are not of interest (for example because they lie outside the required output image).

The procedure by which the method determines which of the elements can be represented by average values in the convolution calculations will now be described.

Firstly, the two root nodes 22 and 24 are compared by calculating the difference between the product of the maximum element values of the nodes 22 and 24 and the product of the minimum element values for the nodes. This is then compared with the threshold TV. If the difference is less than the threshold, only at most two average values, one for each set would be needed to represent the intensity values of all the data elements. In the present example, the threshold is set at an appropriate level between 1 and 9, and is exceeded by said difference, which in this case is 10.

It is then determined whether the data elements represented by the two nodes would only contribute to the portions of the output data set outside the zone of interest (address is 0-15).

This is achieved using the node start positions and lengths, and since the nodes 22 and 24 represent the whole of the data sets 4 and 6, all of the addresses of interest in the output data set 26 will be contributed to by products of data elements represented by the two nodes.

Accordingly, the comparison of the nodes 22 and 24 in this case will determine that intensity values cannot be approximated, nor can combinations be disregarded at this level, and the comparison of nodes proceeds to the next stage.

This involves the comparison of the node 22 with each of the two nodes 32 and 34 in the tree 9 in the same way as discussed above.

In this case, the comparison will determine that neither average values for the elements represented by the node combinations (22 and 32, 22 and 34) can be used, nor would the contributions of the product pairs of those data elements lie wholly outside the addresses of interest in the output set. If this were not the case, however, then the comparison with all of the sub-nodes of the node pair in question need not occur since average values can be used for all of the data elements or of the product contribution of the pairs of those elements do not have to be considered.

Accordingly, if, for example, all of the data elements represented by the node 32 were of zero value, then a comparison of the node 22 (or any of its sub-nodes) with any of the nodes 36, 38, 40, 42, 44 and 46 is not required.

This process continues until the node 22 has been compared against all the necessary nodes in the tree 9 (including the data elements in the set 6).

Each node in the level 1 is then compared with those nodes (starting from the top) of the tree 9 which have not already

been ruled out by the previous comparison(s), and this process is repeated as necessary for all the other nodes in the tree 8, so that each node in the tree 8 is compared against those nodes in the tree 9 which have not already been ruled out from further comparison (as a result of the comparison with higher level nodes in the tree 8).

An alternative threshold test to determine whether average data element values can be used in the convolution of data element groups represented by two given nodes (Node1, Node2) is as follows:

$$\text{pyramidMultiplier} \times (\text{maxNode1} \times \text{maxNode2} - \text{meanNode1} \times \text{meanNode2}) < \text{Threshold1}$$

Where "pyramidMultiplier" is the length (i.e. the number of data elements) of the shorter of the two nodes, or the length of both nodes if they contain the same number of elements, maxNode1/2 is the maximum value of any data element contained in the node and meanNode1/2 is the mean value of the data elements in the node.

If the comparison of a given node with another node or data element determines that the elements represented by said given node can be represented by an average value, it is then determined whether that average value can be approximated to zero.

Using the notation mentioned above, this can be done with the following formula:

$$\text{meanNode1} \times \text{meanNode2} \times \text{pyramidMultiplier} < \text{threshold2}$$

It will be appreciated that, if either Node1 or Node2 is a single data element, its mean will be the value of that data element, whilst the pyramidMultiplier will be 1.

In the present example, threshold2 is 0.12.

However, this approach is not universally correct as it does not consider the worst case scenario in which all the highest value data elements in the node(s) conspire to contribute to the highest valued output contribution. Thus, an alternative approach would be to apply the formula:

$$\text{MaxNode1} \times \text{maxNode2} \times \text{pyramidMultiplier} < \text{threshold}$$

If zero can be used as an average value for the convolution of one group of data elements with another, then that convolution merely involves mapping the zero data values from the convolution of those two groups into the appropriate locations of the output data set.

In the example shown in Figure 2, some of the data elements can be represented by average values, in some cases zero, whilst certain other data elements have to be convolved directly.

For example, the comparison of the nodes 50 and 32 reveals not only that the variance of the data elements represented by those nodes (i.e. the elements at 12 and 13 of the input data set and those at -7 to 0 of the filter set) is below the given threshold, but also that the average for the node 50 is 0 so that the product of the data elements 12 and 13 in the input set with the data elements -7 to 0 in the filter set will always be 0, and it is thus not necessary explicitly to perform the product calculation. Thus, in Figure 3, the convolution product $I(3,6)*F(1,0)$ resulting from the convolution of the nodes 50 and 32 is shown as a series of 0 values which start at address 5 and extend to address 15.

$I(2,0)*F(2,3)$ is a case where the node pair product (for the nodes 20, 52) is approximately 0, i.e. less than the prescribed threshold value. Again, the convolution of the data elements represented by those two nodes does not contribute to the output value of the data set 26.

Convolution $I(2,2)*F(2,0)$ is the case where the values

represented by a node pair 54 and 36 are constant so that they can be represented as single values (average values would also be applied if the data elements contained in each node were not of a constant value, but exhibited sufficiently little variation in value). In this case, the output contribution can be calculated using the same product value for each data element combination. As can be seen from Figure 3, the convolution of those two nodes produces a data set in which the values rise from a minimum of .3 to a peak of 1.2 and then descends back to a minimum of .3. The reason for this variation is that the product of some of the pairs of data elements map into the same addresses in the output data set and therefore accumulate. For example, the elements 58 and 60 provide a product which is mapped into the address 62 as do the elements 64 and 66.

$I(2,3)*F(2,3)$, produces a product which is out of range with respect to the domain of the output data (since the lowest address into which the results of that convolution will be mapped is 17), and this convolution can therefore be disregarded.

$I(4,5)*F(4,7)$ is a case where, due to high local variance in both input and filter nodes, we have to traverse all the way down to the leaf nodes so that the individual data elements need to be convolved in a conventional way.

It will be appreciated that, although the method has been described in relation to one-dimensional data sets, it is also applicable to other data sets, for example data set defining a two-dimensional image and filter.

A description of a further example of a method in accordance with the invention is set out below.

As before, input data are transformed into hierarchical form and the convolution then proceeds by the following Recursive

Decent Procedure (RDP). The RDP is initially executed with the highest node in the first input data set referred to as Node A and the highest node in the second input data set, referred to as Node B, according to the following steps:

1. If the absolute product of the Node Average Element Values of Node A and Node B will produce a total result in the output smaller than the threshold value TV, then in this case the execution of the procedure is complete which has the effect that Node A is not subsequently tested with sub-nodes of Node B and Node B is not subsequently tested with sub-nodes of Node A.
2. If the position and length of Node A and Node B are such that the result of their convolution would only change the intensity of elements which are less than the minimum position in the output data set or greater than the maximum position in the output data set (that is the domain of the output of the convolution would not overlap the domain of the output data set), then in this case the execution of the procedure is complete which has the effect that Node A is not subsequently tested with sub-nodes of Node B and Node B is not subsequently tested with sub-nodes of Node A.
3. If the difference in intensity between the Node A MAXEI and the Node A MINEI is less than TV and the difference in intensity between the Node B MAXEI and the Node B MINEI is less than TV, then in this case Node A and Node B are convolved directly by a suitable known method for the convolution of constant areas, for example as described by the pseudo program code in Appendix 3 using the simplifying assumption that the intensity of the elements in the regions described by Node A and Node B are constant, after having convolved an approximation of the Node A and Node B image data the execution of the procedure is complete which has the effect that Node A is not subsequently tested with sub-nodes of Node B and Node B is not subsequently tested with sub-nodes of Node A.

4. If the Node A Node Length is greater than the Node B Node Length then in this case the RDP is executed first with the lower position sub-node of Node A and Node B, subsequently with the higher position sub-node of Node A and Node B.

5. If the Node B Node Length is greater than the Node A Node Length then in this case the RDP is executed first with the lower position sub-node of Node B and Node A, subsequently with the higher position sub-node of Node B and Node A.

6. If the Node A Node Length and the Node B Node Length are the same and the Node A Node Length is greater than 1 (i.e. Node A is not a sub-element leaf Node) then in this case the RDP is executed first with the lower position sub-node of Node A and Node B subsequently with the higher position sub-node of Node A and Node B.

If neither candidate pair is rejected or the convolution is performed directly, then each candidate is not tested further with sub-nodes of the other candidate. This procedure is achieved by a suitable method, for example as described in the pseudo program code in Appendix 4.

7. Once all invocations of the RDP are complete the output pixel elements have been calculated and the convolution is complete.

A sample implementation of the method according to the invention written in the C programming language is shown in Appendix 1 and the effect of this implementation on two example input lines in Figure 5, 6.1 and 6.2 is shown at 6.3.

An extension of the method in accordance with the invention would be to apply the method to data with higher dimensionalities for example to the convolution of 2D images represented by the known quad-tree representation or to the convolution of data sets with differing dimensionalities, for

example line and 2D image convolution.

A further extension of the method in accordance with the invention would be to test candidate pairs in the convolution for their effect on the image with respect to the output intensity already accumulated in the output image. For example, if the output at a given location in the output image was fully saturated (that is the output intensity has reached some notional maximum value beyond which additional intensity is not usefully represented) then the candidate pair could be rejected at that stage. Similarly the threshold which determines whether a candidate pair is important in the output can be varied depending on the output value already accumulated, so that for example if the output at a given location is already at high intensity convolutions which would provide a relatively small addition in intensity to that location are rejected.

A further extension of the method would allow differing representation of the data at each level in the hierarchy and during a pre-process that representation which produced regions with the lowest variance would be selected. For example, when performing a 2D convolution a two-way sub-division could be used so that each non-element node has two sub-nodes describing two halves of a square, wherein the axis of division (e.g. vertical or horizontal) is chosen to minimise the variance of the intensities described by said sub-nodes.

In a further extension of the method the output image could also be represented hierarchically so that determining the average intensity already accumulated in a region can be done efficiently.

The methods according to the invention place no restriction on the exact size of the data sets and can perform non-periodic convolution without explicit zero-padding of the data. This is generally not the case for Fourier based methods. It is

12-MAR-99 FRI 14:31

KEITH W WASH & Co.

FAX NO. 01223 324353

P.23

19

also worth noting that methods according to the invention are based in the (mathematically) real domain whereas Fourier based methods are inherently based in the complex domain.

ART Ltd, Company Confidential, 1999

Appendix I

```

int number_of_elements = 8;
int number_of_levels = 4;
int element_pointer;

float input_line1[0.0, 0.0, 0.0, 5.0, 4.0, 1.0, 1.0, 1.0];
float input_line2[1.0, 2.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0];
float output_line[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0];

typedef struct node_element {
    int position;
    int length;
    float max_value;
    float min_value;
    float average_value;
    node_element *child1;
    node_element *child2;
}

node_element *make_node(int number, int position) {
    node_element *node;
    node = alloc(sizeof(node_element));
    node->position = position;
    node->length = length;
    if (number == 2) {
        if (input_line[position] > input_line[position+1]) {
            node->max_value = input_line[position];
            node->min_value = input_line[position+1];
        } else {
            node->max_value = input_line[position+1];
            node->min_value = input_line[position];
        }
        node->average = (node->max_value + node->min_value) / 2;
    } else {
        node->child1 = make_node(number/2, position);
        node->child2 = make_node(number/2, position + number/2);
    }
    return (node);
}

get_stats(node_element *node) {
    if (node->child1 != NULL) {
        get_stats(node->child1);
        get_stats(node->child2);
        node->average = (node->child1->average + node->child2->average) / 2.0;
        if (node->child1->max_value > node->child2->max_value) {
            node->max_value = node->child1->max_value;
        } else {
            node->max_value = node->child2->max_value;
        }
        if (node->child1->min_value < node->child2->min_value) {
            node->min_value = node->child1->min_value;
        } else {
            node->min_value = node->child2->min_value;
        }
    }
    return();
}
return();

```


ART Ltd. Company Confidential, 1999

```

initialise_structures(node_element *top_node) {
    top_node = make_nodes(number_of_elements, 0);
    get_stats(top_node);
}

void convolve(node_element *node1, node_element *node2) {
    int
        iStart,
        iEnd,
        i,
        /* demarkation of resultant convolution regions */
        spanLen = node1->length + node2->length - 1,
        constLen = abs(node1->length - node2->length) + 1,
        slopeLen = (spanLen - constLen)>>1,
        xConvStart = node1->position + node2->position,
        xConvEnd = xConvStart + spanLen;

    float
        valProduct = node1->average_value * node2->average_value;

    overlapLen = 0; /* zero to catch none-overlap cases */
    if(xConvStart < 0) {
        if(xConvEnd >= numElements)
            overlapLen = numElements;
        else
            overlapLen = xConvEnd;
    } else {
        if(xConvEnd >= numElements)
            overlapLen = numElements - xConvStart;
        else
            overlapLen = spanLen;
    }

    /* accumulate left slope */
    iStart = (xConvStart < 0) ? 0 : xConvStart;
    iEnd = xConvStart + slopeLen > numElements ? numElements : xConvStart +
slopeLen;
    for(i=iStart; i<iEnd; i++);
        output_line[i] += (float)(i - xConvStart)*valProduct;

    /* accumulate constant section */
    iStart = (xConvStart + slopeLen < 0) ? 0 : xConvStart + slopeLen;
    iEnd = xConvEnd - slopeLen > numElements ? numElements : xConvEnd - slopeLen;
    for(i=iStart; i<iEnd; i++);
        output_line[i] += (float)(slopeLen + 1)*valProduct;

    /* accumulate right slope */
    iStart = (xConvEnd - slopeLen < 0) ? 0 : xConvEnd - slopeLen;
    iEnd = xConvEnd > numElements ? numElements : xConvEnd - slopeLen;
    for(i=iStart; i<iEnd; i++);
        output_line[i] += (float)(xConvEnd - i)*valProduct;
}

test_candidates(node1, node2) {
    /* test1 return if the absolute level of intensity at this level is low */
    if ((node1->average * node2->average) < average_threshold)
        return();

    /* test2 return if the result falls outside the domain of the output */
    if ((node1->position - node2->position - half_length) > number_of_elements)
        return();
}

```

ART Ltd, Company Confidential, 1999

```
if ((node1->position + node1->length + node2->position + node2->length -
half_length) < 0)
    return();

/* test3 convolve directly if the variance is low */
if (node1->min_value < node2->min_value) {
    min_value = node1->min_value;
} else {
    min_value = node2->min_value;
}
if (node1->max_value > node2->max_value) {
    max_value = node1->max_value;
} else {
    max_value = node2->max_value;
}
if ((max_value - min_value) > variance_threshold) {
    convolve(node1, node2);
    return();
}
if (node1->child1 != NULL) {
    test_candidates(node1->child1, node2);
    test_candidates(node1->child2, node2);
}
if (node2->child1 != NULL) {
    test_candidates(node1, node2->child1);
    test_candidates(node1, node2->child2);
}
return();
}

main() {
    initialise_structures(top_node1);
    initialise_structures(top_node2);
    test_candidates(top_node1, top_node2);
}
```

ART Ltd, Company Confidential, 1999

APPENDIX 2 (Hierarchy for Fig. 1)

```
get_stats(node_element *node) {
    if (node->child1 != NULL) {
        get_stats(node->child1);
        get_stats(node->child2);
        node->average = (node->child1->average + node->child2->average) / 2.0;
        if (node->child1->max_value > node->child2->max_value) {
            node->max_value = node->child1->max_value;
        } else {
            node->max_value = node->child2->max_value;
        }
        if (node->child1->min_value < node->child2->min_value) {
            node->min_value = node->child1->min_value;
        } else {
            node->min_value = node->child2->min_value;
        }
    }
    return();
}

return();

}

initialise_structures(node_element *top_node) {
    top_node = make_nodes(number_of_elements, 0);
    get_stats(top_node);
}

initialise_structures(top_node1);
initialise_structures(top_node2);
```

ART Ltd. Company Confidential 1999

APPENDIX 3 Example constant convolution

```

void convolve(node_element *node1, node_element *node2) {
    int
        iStart,
        iEnd,
        i;
    /* demarkation of resultant convolution regions */
    spanLen = node1->length + node2->length - 1,
    constLen = abs(node1->length - node2->length) + 1,
    slopeLen = (spanLen - constLen)>>1,
    xConvStart = node1->position + node2->position,
    xConvEnd = xConvStart + spanLen;

    float
        valProduct = node1->average_value * node2->average_value;

    overlapLen = 0; /* zero to catch none-overlap cases */
    if(xConvStart < 0) {
        if(xConvEnd >= numElements)
            overlapLen = numElements;
        else
            overlapLen = xConvEnd;
    } else {
        if(xConvEnd >= numElements)
            overlapLen = numElements - xConvStart;
        else
            overlapLen = spanLen;
    }

    /* accumulate left slope */
    iStart = (xConvStart < 0) ? 0 : xConvStart;
    iEnd = xConvStart + slopeLen > numElements ? numElements : xConvStart +
slopeLen;
    for(i=iStart; i<iEnd; i++);
        output_line[i] += (float)(i - xConvStart)*valProduct;

    /* accumulate constant section */
    iStart = (xConvStart + slopeLen < 0) ? 0 : xConvStart + slopeLen;
    iEnd = xConvEnd - slopeLen > numElements ? numElements : xConvEnd - slopeLen;
    for(i=iStart; i<iEnd; i++);
        output_line[i] += (float)(slopeLen + 1)*valProduct;

    /* accumulate right slope */
    iStart = (xConvEnd - slopeLen < 0) ? 0 : xConvEnd - slopeLen;
    iEnd = xConvEnd > numElements ? numElements : xConvEnd - slopeLen;
    for(i=iStart; i<iEnd; i++);
        output_line[i] += (float)(xConvEnd - i)*valProduct;
}

```

ART Ltd, Company Confidential, 1999

APPENDIX 4 If either candidate pair is rejected or convolution is performed directly then each candidate is not tested further with sub-nodes of other candidates

```
test_candidates(node1, node2) {  
    /* test1 return if the absolute level of intensity at this level is low */  
    if ((node1->average * node2->average) < average_threshold)  
        return();  
  
    /* test2 return if the result falls outside the domain of the output */  
    if ((node1->position - node2->position - half_length) > number_of_elements)  
        return();  
    if ((node1->position + node1->length + node2->position + node2->length -  
half_length) < 0)  
        return();  
  
    /* test3 convolve directly if the variance is low */  
    if (node1->min_value < node2->min_value) {  
        min_value = node1->min_value;  
    } else {  
        min_value = node2->min_value;  
    }  
    if (node1->max_value > node2->max_value) {  
        max_value = node1->max_value;  
    } else {  
        max_value = node2->max_value;  
    }  
    if ((max_value - min_value) > variance_threshold) {  
        convolve(node1, node2);  
        return();  
    }  
    if (node1->child1 != NULL) {  
        test_candidates(node1->child1, node2);  
        test_candidates(node1->child2, node2);  
    }  
    if (node2->child1 != NULL) {  
        test_candidates(node1, node2->child1);  
        test_candidates(node1, node2->child2);  
    }  
    return();  
}
```

ART Ltd, Company Confidential, 1999

Fig 1

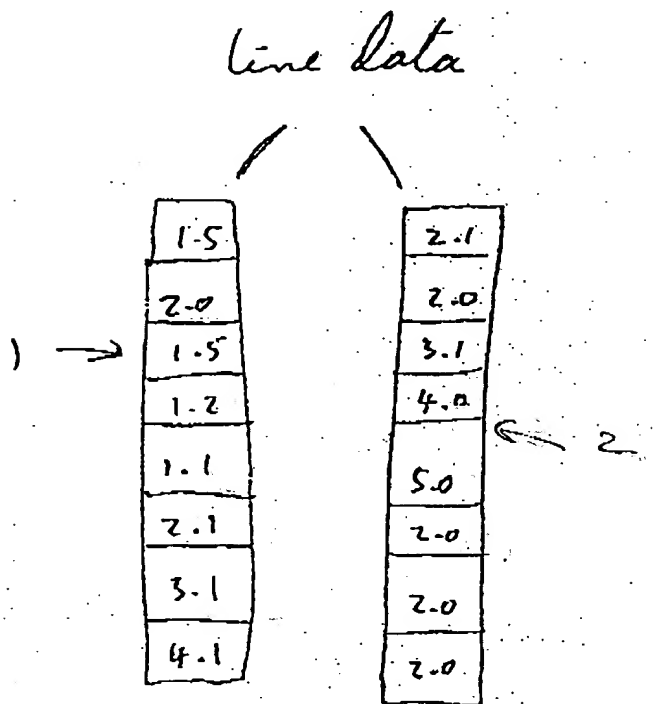
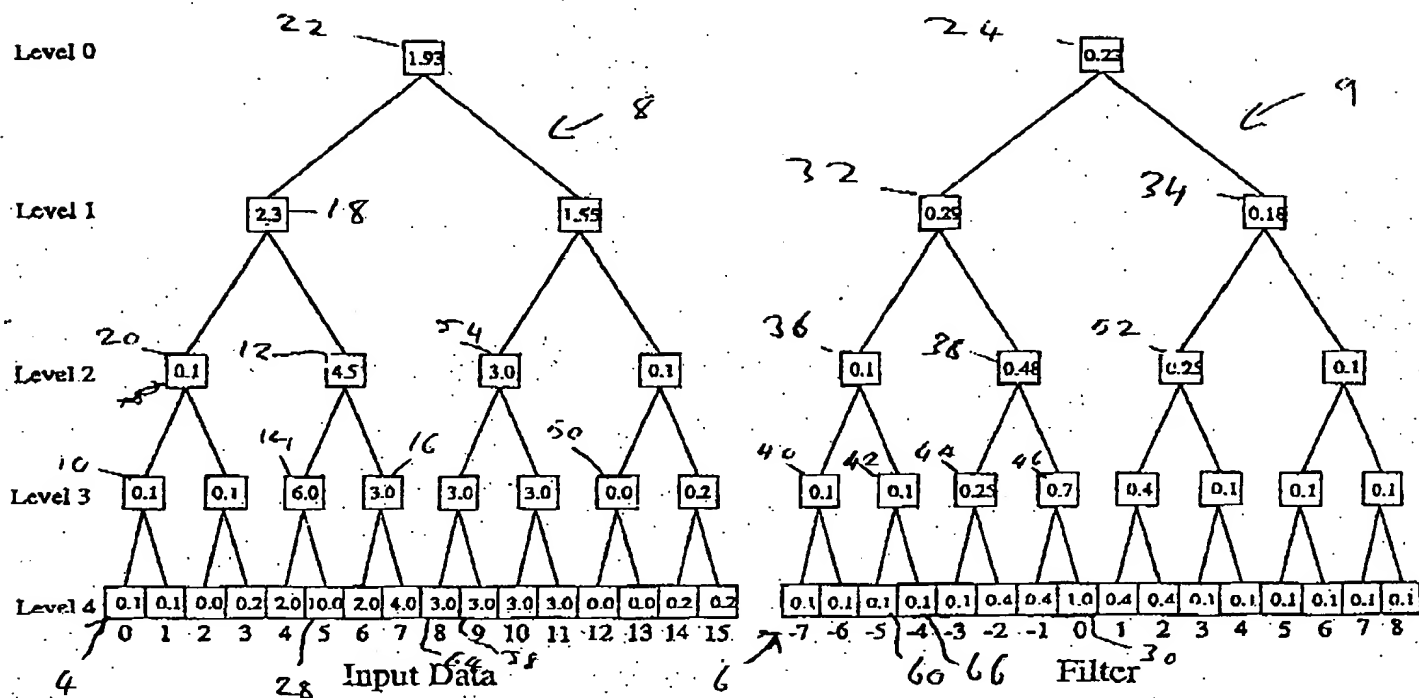
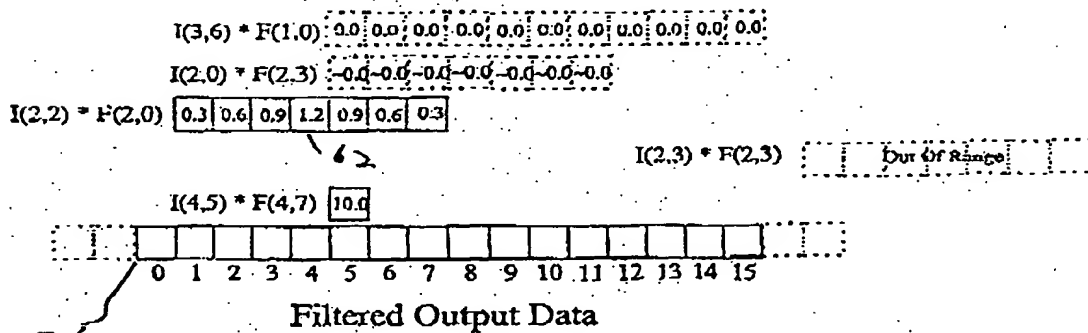


Fig. 1

Fig 2 Hierarchical Convolution



Node-pair product contributions



ART Ltd, Company Confidential, 1999

Fig 3

Data node

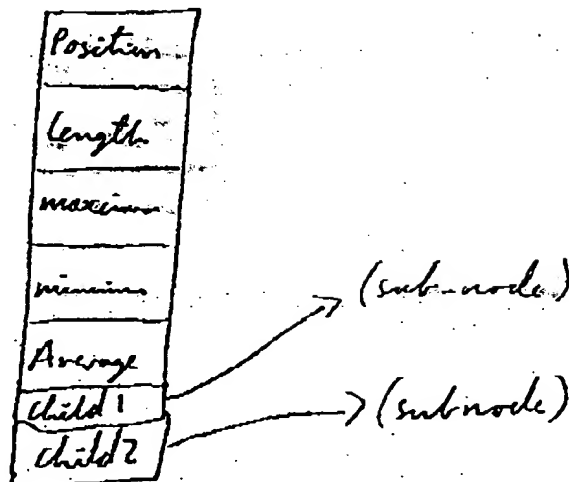


Fig. 4

12-MAR-99 FRI 14:33

KEITH W. WASH & Co.

FAX NO. 01223 324353

P. 33

ART Ltd, Company Confidential, 1999

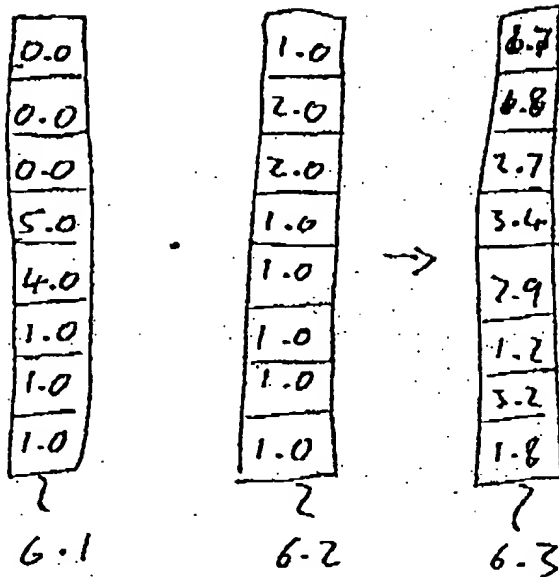
~~Fig. 6~~

Fig. 5

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☒ OTHER: Black Spots

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)